



21 世纪高等学校规划教材
Textbook Series of 21st Century

C 语言程序设计

许薇薇 编



中国电力出版社

<http://jc.cepp.com.cn>



21 世纪高等学校规划教材
Textbook Series of 21st Century

C 语言程序设计

许薇薇 编
李 珍 主审



中国电力出版社

<http://jc.cepp.com.cn>

内 容 提 要

本书为 21 世纪高等学校规划教材。本书是学习 C 语言程序设计的基础教程，书中全面地介绍了 C 语言的基本概念、数据类型、语句及结构特点，系统地讲述了 C 语言程序设计的基本方法和技巧。

本书采取循序渐进的内容安排，通俗易懂的讲解方法，并辅以大量的例题，讲述力求理论联系实际、深入浅出；注重培养读者的程序设计能力及良好的程序设计风格和习惯；注重实践环节，精选了较多的习题和上机操作题。

本书可作为普通高等学校计算机专业和非计算机专业 C 语言程序设计课程的本、专科教材（可以根据本科、专科教学要求的不同进行适当取舍），也可供计算机培训班或其他自学者使用。

图书在版编目 (CIP) 数据

C 语言程序设计/许薇薇编. —北京：中国电力出版社，
2006

21 世纪高等学校规划教材

ISBN 7 - 5083 - 3787 - 5

I. C... II. 许... III. C 语言—程序设计—高等学
校—教材 IV. TP312

中国版本图书馆 CIP 数据核字 (2005) 第 156784 号

中国电力出版社出版、发行

(北京三里河路 6 号 100044 <http://jc.cepp.com.cn>)

北京密云红光印刷厂印刷

各地新华书店经售

*

2006 年 1 月第一版 2006 年 1 月北京第一次印刷
787 毫米 × 1092 毫米 16 开本 16.25 印张 370 千字
印数 0001—4000 册 定价 24.80 元

版 权 专 有 翻 印 必 究

(本书如有印装质量问题，我社发行部负责退换)

前 言

随着计算机技术的飞速发展,计算机已在各行各业得到了广泛应用。计算机技术的中心是程序设计,掌握一定的程序设计技术,是当今社会人们应当必备的技能与文化素质之一。掌握程序设计的前提是掌握程序设计语言,在众多的程序设计语言中,C 语言以其灵活性和实用性受到编程人员的广泛青睐。

C 语言概念简洁,数据类型丰富,表达能力强,运算符多而灵活,程序结构性和可读性好,易于体现结构化程序设计思想。它既具有高级语言的特点,又具有汇编语言的功能;既能有效地进行算法描述,又能对硬件直接进行操作;既适合编写应用程序,又适合开发系统软件,它是当今世界上应用广泛、最具影响的程序设计语言之一。此外,C 语言还具有整个语言紧凑整齐、设计精巧、编辑方便、编译与目标代码运行效率高、操作简便、使用灵活等许多鲜明的特点。

本书全面介绍了 C 语言的概念、特性和结构化程序设计方法。全书共有 12 章,第 1 章介绍了 C 语言程序设计的基本知识。第 2 章介绍了 C 语言的基本数据类型、常量、变量以及表达式。第 3、4、5 章介绍了 C 语言进行结构化程序设计的基本方法,包括数据输入输出、顺序结构、选择结构、循环结构及其设计方法。第 6 章介绍了函数与预编译处理。第 7、8、9、10 章对 C 语言的数组、指针、结构体、共用体等作了较详细的介绍。第 11 章对 C 语言的文件操作做了详细的阐述。第 12 章介绍了 C 语言综合编程应用。本书在各章后都配有一定的上机操作题,意在加强实践环节,通过实际练习更好地掌握 C 语言。

在本书的编写过程中,得到了李卓玲教授的热情支持与指导并提出许多宝贵意见。本书由李珍教授担任主审,在审阅过程中提出了很多极有价值的意见和建议。在此对两位老师表示衷心的感谢。

由于编者水平有限,书中难免有错误和不当之处,敬请读者批评指正。

编 者

2005 年 9 月

目 录

前言

第 1 章 C 语言概述	1
1.1 C 语言发展概述	1
1.2 C 语言的特点	2
1.3 简单的 C 语言程序	2
1.4 C 语言程序的开发环境	3
习题	6
上机操作	6
第 2 章 基本数据类型及表达式	8
2.1 C 语言的数据类型	8
2.2 常量和变量	8
2.3 整型数据	9
2.4 字符型数据	10
2.5 实型数据	11
2.6 变量赋初值	12
2.7 运算符及表达式	12
2.8 数据类型转换	16
习题	17
上机操作	18
第 3 章 顺序结构程序设计	20
3.1 C 语句概述	20
3.2 数据输入与输出	22
3.3 程序举例	29
习题	31
上机操作	31
第 4 章 选择结构程序设计	33
4.1 关系运算符和关系表达式	33
4.2 逻辑运算符与逻辑表达式	34
4.3 if 语句	35
4.4 switch 语句	41
4.5 程序举例	43
习题	46
上机操作	47

第 5 章 循环结构程序设计	49
5.1 while 语句	49
5.2 do-while 语句	50
5.3 for 语句	52
5.4 break 语句、continue 语句和 goto 语句	55
5.5 循环的嵌套	57
5.6 程序举例	59
习题	63
上机操作	64
第 6 章 函数与编译预处理	66
6.1 模块化程序设计与函数	66
6.2 函数的定义与调用	67
6.3 函数的递归调用	72
6.4 变量作用域与存储方式	77
6.5 编译预处理	85
6.6 程序举例	89
习题	91
上机操作	91
第 7 章 数组	94
7.1 一维数组的定义和引用	94
7.2 二维数组的定义和引用	96
7.3 字符数组与字符串	99
7.4 数组作为函数的参数	107
7.5 程序举例	111
习题	117
上机操作	118
第 8 章 指针	121
8.1 指针与指针变量	121
8.2 指针与函数	127
8.3 指针与数组	134
8.4 指针与字符串	141
8.5 指针数组与命令行参数	145
8.6 程序举例	149
习题	151
上机操作	152
第 9 章 结构体类型与链表	155
9.1 结构体类型的定义	155
9.2 结构体类型变量	157

9.3 结构体类型数组	160
9.4 指向结构体的指针	162
9.5 结构体与函数	164
9.6 链表	167
9.7 程序举例	178
习题	181
上机操作	182
第 10 章 共用体与枚举类型	184
10.1 共用体类型与共用体变量	184
10.2 共用体变量的引用	185
10.3 共用体变量的应用	187
10.4 位运算	189
10.5 位段结构	192
10.6 枚举类型	194
10.7 用 typedef 定义类型	196
习题	198
上机操作	199
第 11 章 文件	201
11.1 文件概述	201
11.2 文件类型指针	203
11.3 文件的打开与关闭	204
11.4 文件的读写	207
11.5 文件的定位与随机读写	216
11.6 文件的出错检测	218
习题	218
上机操作	219
第 12 章 C 语言综合应用程序示例	221
12.1 简单的数据库管理	221
12.2 用 C 语言编制画图程序	224
附录 I 常用字符与 ASCII 代码对照表	232
附录 II C 语言中的关键字	234
附录 III 运算符和结合性	234
附录 IV Turbo C 2.0 编译、连接时常见的错误和警告信息	236
附录 V C 常用部分库函数	241
参考文献	249

第 1 章 C 语言概述

教学目标

了解 C 语言程序的基本结构, 掌握 C 语言程序的上机步骤。

本章要点

- C 语言的特点
- C 语言程序的构成
- C 语言程序的上机步骤
- C 语言程序的调试方法

1.1 C 语言发展概述

C 语言是国际上广泛流行的一种计算机高级语言。用 C 语言既可以编写系统软件, 也可以编写应用软件。

C 语言是在 1972~1973 年间由美国贝尔实验室的 D. M. Ritchie 和 K. Thompson 以及英国剑桥大学的 M. Richards 等为描述和实现 UNIX 操作系统而设计的。UNIX 操作系统源代码的 90% 以上是用 C 语言编写的。UNIX 操作系统的一些主要特点, 如便于理解、易于修改、具有良好的可移植性等, 在一定程度上都受益于 C 语言。所以, UNIX 操作系统的成功与 C 语言是密不可分的。

最初的 C 语言附属于 UNIX 的操作系统环境, 而它的产生却可以更好地描述 UNIX 操作系统。时至今日, C 语言已独立于 UNIX 操作系统。它已成为微型、小型、中型、大型和超大型 (巨型) 计算机上通用的一种程序设计语言。D. M. Ritchie 和 K. Thompson 也以他们在 C 语言和 UNIX 系统方面的卓越贡献获得了很高的荣誉。1982 年, 他们获得了《美国电子学杂志》颁发的成就奖, 成为该奖自颁发以来首次因软件工程成就而获奖的获奖者。1983 年, 他们又获得了计算机界的最高荣誉奖——图灵奖。

随着 C 语言的不断发展、应用和普及, 目前, C 语言已经能够在多种操作系统下运行。实用的 C 语言编译系统种类繁多, 如 Microsoft C、Turbo C 等。C 语言是近年来最受人们欢迎的一种系统程序设计语言之一。

随着计算机应用领域的不断扩展和深入, 作为人与计算机进行信息交流工具之一的 C 程序设计语言同样得到了迅速的发展。C 语言从最初的只是为描述和实现 UNIX 操作系统而提出的一种程序设计语言; 后来作为风靡全球的面向过程的计算机程序设计语言, 用在大、中、小及微型机上; 目前正在流行的是面向对象的程序设计语言 C++。

C++ 是在 C 语言的基础上发展起来的程序设计语言。我们不仅可以用其编写面向对象的程序, 也可以用其编写面向过程的程序。C++ 等面向对象的程序设计语言是开发

Windows (图形界面操作系统) 环境应用程序的编程语言。

1.2 C 语言 的 特点

C 语言能够成为目前应用广泛的高级程序设计语言之一,完全是由其语言特点决定的。C 语言的特点可大致归纳如下:

(1) 语言基本组成部分紧凑简洁。C 语言只有 32 个标准关键字, 34 个标准运算符以及 9 条控制语句, 不但语言的组成精练、简洁, 而且使用方便、灵活。

(2) C 语言运算符丰富, 表达能力强。C 语言具有“高级语言”和“低级语言”的双重特点, 其运算符包含的内容广泛, 所生成的表达式简练、灵活, 有利于提高编译效率和目标代码的质量。

(3) C 语言数据结构丰富, 结构化强。C 语言提供了编写结构化程序所需要的各种数据结构和控制结构, 这些丰富的数据结构和控制结构及以函数调用为主的程序设计风格, 保证了利用 C 语言所编写的程序能够具有良好的结构化。

(4) C 语言具有结构化的控制语句, 如 if-else 语句、switch 语句、while 语句、do-while 语句、for 语句。C 语言用函数作为程序模块以实现程序的模块化, 是结构化的理想语言, 符合现代编程风格。

(5) C 语言提供了某些接近汇编语言的功能, 有利于编写系统软件。C 语言提供的一些运算和操作, 能够实现汇编语言的一些功能, 如它可以直接访问物理地址, 并能进行二进制位运算等, 这为编写系统软件提供了方便条件。

(6) C 语言程序所生成的目标代码质量高。C 语言程序所生成的目标代码效率仅比用汇编语言描述同一个问题低 20% 左右。

(7) C 语言程序可移植性好。C 语言所提供的语句中, 没有直接依赖于硬件的语句, 与硬件有关的操作, 如数据的输入、输出等都是通过调用系统的库函数来实现的, 而库函数本身不是 C 语言的组成部分。因此用 C 语言编写的程序可以很容易地从一种计算机环境移植到另一种计算机环境中。

C 语言的弱点: 一是运算符的优先级较复杂, 不容易记忆; 二是由于 C 语言的语法限制不太严格, 在增强程序设计灵活性的同时, 也在一定程度上降低了某些安全性, 因此对程序设计人员提出了更高的要求。

1.3 简单的 C 语言程序

【例 1.1】 从键盘上读入两个整数, 计算这两个整数之和并显示出来。

```
#include "stdio.h"

main()
{ int a,b,sum;
```

```
printf("Enter two numbers: ");
scanf("%d,%d",&a,&b);
sum=a+b;
printf("The sum is %d\n",sum);
}
```

【例 1.2】 从键盘上读入两个整数，求出其中最大的数并显示。

```
#include "stdio.h"
main()
{ int a,b,c;
  printf("Enter two numbers: ");
  scanf("%d%d",&a,&b);
  c=max(a,b);
  printf("The max is %d\n",c);
}

int max(int x,int y)      /* 函数定义 */
{ int z;
  if(x>y)z=x;
  else z=y;
  return z;
}
```

从以上的程序例子中可以看出 C 语言程序的构成规则：

- (1) C 语言是由函数构成的，其中至少（也只能）包括一个主函数 main（）。
- (2) 函数体是由花括号 {} 括起来的部分。
- (3) C 语言中的每个语句都以“;”结束。
- (4) C 语言书写格式自由，一行内可以写一个语句，也可以写多个语句。
- (5) #include 是编译预处理命令，其作用是将由双引号或尖括号括起来的文件内容读入该命令位置处。对于输入输出库函数一般需要使用 #include 命令将“stdio.h”文件包含到源文件中。#include 命令的使用方法将在第 6 章介绍。
- (6) 可用 /* ... */ 对 C 语言程序中的任何部分作注释。
- (7) C 语言中所有变量都必须先定义类型，然后再使用。
- (8) 一个 C 语言程序通过函数之间的相互调用来实现相应的功能。函数既可以是系统提供的库函数（标准函数），也可以是根据问题的需要自己定义的函数。

1.4 C 语言程序的开发环境

一个编写好的 C 语言源程序要经过编辑（输入）、编译和连接后才能形成可执行的程序。C 语言的编译系统有多种，本书采用 Turbo C 2.0 集成开发环境并介绍在此环境下如何编辑、编译、连接和运行 C 程序。

1.4.1 C 程序的上机步骤

1. 编辑

在 Turbo C 环境下, 将 C 语言源程序通过键盘输入到计算机中, 并以文件形式存盘, C 语言源程序文件的扩展名是 .C。

2. 编译

将源程序输入到计算机并以文件形式存盘后, 需要经过 C 语言编译器将其转换成目标程序。转换后的目标文件的扩展名是 .OBJ。在编译过程中, 若发现源程序中有语法错误, 则需要对其进行修改, 然后重新编译。

3. 连接

经过编译后生成的目标文件不能直接执行, 需要经过连接之后才能生成可执行代码。在 DOS 系统下, 连接后所得到的可执行文件扩展名是 .EXE。

所谓连接, 是将目标文件和库函数或其他目标程序进行连接, 生成可执行的目标程序。

4. 执行

连接后源程序的目标程序就是可执行文件了。在 DOS 系统下, 只要键入可执行文件名, 按回车键就可以执行程序了。

综上所述, 上机步骤为:

编辑(.C)——编译(.OBJ)——连接(.EXE)——执行。

1.4.2 Turbo C 2.0 介绍

Turbo C 2.0 不仅是一个快速、高效的编译软件, 同时还带有一个易学、好用的集成环境。使用 Turbo C 2.0 时程序的编辑、编译、连接和运行所有这些功能都组合在 Turbo C 2.0 的集成环境内, 并且可以通过菜单选择使用这些功能。

1. Turbo C 2.0 集成开发环境的主屏幕

Turbo C 2.0 系统是 DOS 环境下的系统软件。在 DOS 系统下进入 Turbo C 2.0 所在的目录, 输入 TC, 回车 (安装 Turbo C 2.0 时默认设置) 或在 Windows 系统下进入 Turbo C 2.0 所在文件夹, 双击 TC 文件图标, 即进入 Turbo C 2.0 集成环境。按任意键, 版本信息消失, 屏幕显示如图 1-1 所示。

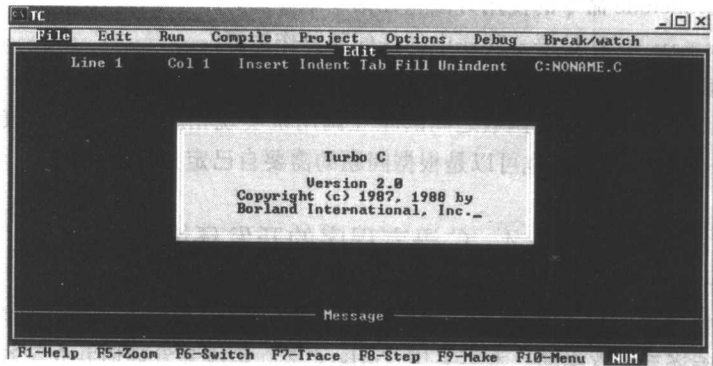


图 1-1 Turbo C 2.0 集成开发环境主屏幕

Turbo C 2.0 集成开发环境的主屏幕，由上至下分成四部分：主菜单、编辑窗口、信息窗口和默认的快速参考行。

2. 主菜单功能

Turbo C 2.0 的集成开发环境的主菜单一共有 8 个选择项，其功能如表 1-1 所示。

表 1-1 主菜单的项目和功能表

项 目	功 能
File	调入和保存文件、管理目录、调用 DOS 和退出 Turbo C 环境
Edit	进入编辑状态，用户可编辑和修改当前编辑窗口中的源程序
Run	控制程序的运行方式，编译、连接和运行当前程序
Compile	编译当前环境内的程序，生成目标文件及可执行文件
Project	处理由多个源程序文件组成的工程文件
Options	设置有关编译和连接的选项
Debug	调试程序，显示变量的值，查找函数，查看调用栈的状态
Break/Watch	调试程序，断点的设置和清除及观察变量值的变化

3. 常用菜单简介

主菜单下的各子菜单可用光标键移动进行选择，回车则执行。也可用每一项的第一个大写字母直接选择。按 Esc 键可从下一级菜单退回。

(1) File 下拉菜单项。File 下拉菜单项如图 1-2 所示，共有 9 个子菜单，子菜单后面是相应的热键，常用子菜单的功能如下。

Load: 将已存在的文件调入编辑窗口；

Pick: 在文件列表选择一个或几个不同的文件装入编辑窗口；

New: 建立一个新文件，此文件名暂时是 NONAME.C；

Save: 将编辑窗口中的文件存盘；

Write to: 将编辑窗口中的文件另存为其他文件名；

Directory: 显示当前目录中的文件列表；

Change dir: 显示当前目录，并允许改变当前驱动器及目录；

OS shell: 暂时退出 Turbo C，进入 DOS 状态，输入 EXIT 并按回车键，返回 Turbo C；

Quit: 退出 Turbo C，返回 DOS 状态。

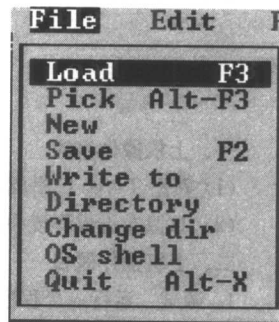


图 1-2 File 下拉菜单项

(2) Edit 菜单项。Edit 菜单项没有下拉子菜单，选择该菜单进入编辑状态，可编辑和修改当前编辑窗口中的源程序。

(3) Run 下拉菜单项。Run 下拉菜单项如图 1-3 所示，其中常用子菜单的功能如下。

Run: 运行程序直到程序结束或下一个断点处；

User screen: 观察程序运行时的屏幕显示，按任意键返回集成环境。

(4) Compile 下拉菜单项。Compile 下拉菜单项如图 1-4 所示, 其中常用子菜单的功能如下。

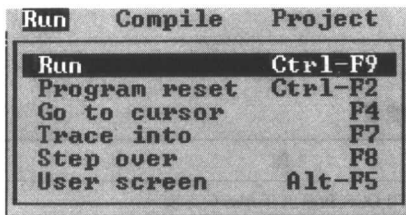


图 1-3 Run 下拉菜单项

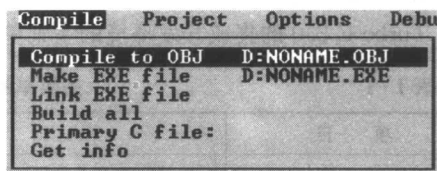


图 1-4 Compile 下拉菜单项

Compile to OBJ: 编译生成目标代码;

Make EXE file: 生成可执行文件;

Link EXE file: 连接当前的 OBJ 和库文件生成 EXE 文件。

习 题

- 1.1 概述 C 语言的主要特点及用途。
- 1.2 构成 C 语言程序的基本单位是什么? 它由哪几部分组成?
- 1.3 简述 C 语言程序调试、运行的步骤。
- 1.4 上机运行本章的两个例题。

上 机 操 作

一、上机操作目的

- (1) 熟悉 Turbo C 集成环境。
- (2) 掌握在集成环境中编辑、编译、连接和运行 C 程序的方法。

二、上机操作题目

1. 调试、运行一个程序

调试、运行一个程序可按以下步骤进行:

- (1) 启动 Turbo C 集成环境, 输入以下程序:

```
main()
{
    int a,b,sum;
    printf("Enter a,b: ");
    scanf("%d,%d",&a,&b);
    sum=a+b;
    printf("sum=%d\n",sum);
}
```

- (2) 存盘。按 F2 键 (或按 Alt+F 打开 File 菜单, 选 Save 命令项), 在弹出的 “Rename NONAME” 框中给上面输入的程序起一个文件名 (即修改默认的文件名 NONAME.C)。

- (3) 编译。按 Alt+C 打开 Compile 菜单，选 Compile to OBJ 命令项。
- (4) 若编译有错，改正程序中的语法错误，重新编译。否则，进入步骤 (5)。
- (5) 连接。按 Alt+C 打开 Compile 菜单，选 Link EXE file 命令项。
- (6) 若连接有错，改正程序中库函数调用错误，重新编译、连接。否则，进入步骤 (7)。
- (7) 运行。按 Alt+R 打开 Run 菜单，选 Run 命令项。
- (8) 按 Alt+F5 观察输出结果。

重复步骤 (7)、(8)，多次运行程序，输入多组数值，观察输出结果。

2. 调试、运行例 1.2 程序

调试、运行 [例 1.2] 程序可按以下步骤进行：按 Alt+F 打开 File 菜单，选 New 命令项（开始编辑一个新文件），输入例 1.2 程序。接下来的操作与调试、运行程序相同，即顺序完成 (2) ~ (8)。

第 2 章 基本数据类型及表达式

教学目标

掌握 C 语言基本数据类型及变量定义的方法, 掌握 C 语言运算符及由各类运算符构成的表达式。

本章要点

- 数据类型及变量定义
- 运算符及表达式

2.1 C 语言的数据类型

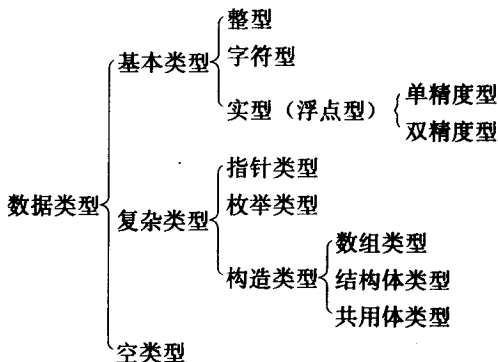
我们学习 C 语言, 是要用 C 语言来编写程序, 通过程序的运行让计算机解决实际问题。一个程序应包括以下两方面内容:

- (1) 对数据的描述。在程序中要指定数据的组织形式, 即数据结构。
- (2) 对操作的描述, 即操作步骤, 也就是算法。

因此, 著名计算机科学家沃思 (Nikiklaus Wirth) 提出一个公式:

$$\text{数据结构} + \text{算法} = \text{程序}$$

C 语言的数据结构是以数据类型形式出现的。C 语言的数据类型如下:



C 语言中数据有常量和变量之分, 它们分别属于以上这些类型。

2.2 常量和变量

2.2.1 常量

常量又称常数, 是指在程序运行中其值不能被改变的量。常量可分为不同的类型, 如

整型常量、实型常量和字符型常量等。可以用一个名字（字符序列）来代表一个常量，这种常量被称为符号常量，例如下面用 `#define` 命令行定义的 `PI` 代表常量 3.14159。

```
#define PI 3.14159
```

有关 `#define` 命令行的详细用法参见第6章。

2.2.2 变量

变量是指在程序运行过程中其值可以被改变的量。一个变量应该有一个名字并属于某一种数据类型，在内存中占据一定的存储单元，用来存放该变量的值。

用来标识变量名、符号常量名、函数名、数组名、类型名、文件名的有效字符序列称为标识符，标识符就是一个名字。

C语言规定标识符只能由英文字母、数字、下划线组成，且第一个字符必须是字母或下划线，一般不超过8个字符；标识符中字母大小写的含义不同；不能使用C语言中的关键字做标识符，例如 `int`；在使用标识符时，应当尽量遵循“简洁明了”和“见名知意”的原则。

变量在使用之前必须先进行定义，即“先定义，后使用”。变量定义的一般形式为

<类型说明符> <变量名表>

类型说明符决定了变量的取值范围和占用内存空间的字节数，变量名表是具有同一数据类型变量的集合。例如：

```
int a,b,c;           /* 定义 a,b,c 为整型变量 */  
float x,y;           /* 定义 x,y 为单精度型变量 */
```

上述变量被定义后，根据其类型的不同，应有不同大小的存储空间。

2.3 整型数据

2.3.1 整型常量

C语言中整型常量可用以下三种形式表示：

- (1) 十进制整数，直接以数字开头的十进制数，例如 123，-234，0。
- (2) 八进制整数，是以 0 开头，例如 065 表示八进制数 65，即 $(65)_8$ 。
- (3) 十六进制整数，是以 0x 开头，例如 0xab 表示十六进制数 ab，即 $(ab)_{16}$ 。

2.3.2 整型变量

整型变量可用来存放整型数据（即不带小数点的数）。整型变量可以分为基本型、短整型、长整型和无符号型 4 种类型，其定义时类型说明符如下。

- (1) 基本型：用 `int` 表示。
- (2) 短整型：用 `short int` 或 `short` 表示。
- (3) 长整型：用 `long int` 或 `long` 表示。
- (4) 无符号型。
 - 1) 无符号整型：用 `unsigned int` 或 `unsigned` 表示；
 - 2) 无符号短整型：用 `unsigned short int` 或 `unsigned short` 表示；
 - 3) 无符号长整型：用 `unsigned long int` 或 `unsigned long` 表示。

无符号整型变量将存储单元中所有二进制位都用来存放数据本身，它没有符号位，即不能存放负数。一个无符号整型变量的取值范围刚好是其有符号数表示范围上下界绝对值之和。

C 标准没有具体规定以上各类数据所占据内存的字节数，各种编译系统在处理上有所不同。一般的原则是，以一个机器字（word）存放一个 int 型数据，long 型数据的字节数应不小于 int 型数据，而 short 型数据应不长于 int 型数据。以 PC 机为例，整型数据所占位数及数的范围如表 2-1 所示。

表 2-1 整型数据所占位数及数的范围

数据类型	所占位数	所占字节数	数的范围
int	16	2	$-32768 \sim 32767$ ，即 $-2^{15} \sim 2^{15} - 1$
unsigned	16	2	$0 \sim 65535$ ，即 $0 \sim 2^{16} - 1$
short	16	2	$-32768 \sim 32767$ ，即 $-2^{15} \sim 2^{15} - 1$
unsigned short	16	2	$0 \sim 65535$ ，即 $0 \sim 2^{16} - 1$
long	32	4	$-2147483648 \sim 2147483647$ ，即 $-2^{31} \sim 2^{31} - 1$
unsigned long	32	4	$0 \sim 4294967295$ ，即 $0 \sim 2^{32} - 1$

2.3.3 整型变量的使用

整型变量应根据要存放数据的取值范围，将其定义成不同的类型。如果一个整型变量取值在 $-32768 \sim 32767$ 之间，应该定义为 int 型。对于不可能有负值的整型变量，应该定义为 unsigned 型。当整型变量的取值可能超出 $-32768 \sim 32767$ 或 $0 \sim 65535$ 的范围时，应定义为 long 型或 unsigned long 型。

2.4 字符型数据

2.4.1 字符型常量

字符型常量是用单引号括起来的单个字符，例如，'A'、'a'、'2'、'%'等，都是有效的字符

表 2-2 转义字符及其含义

字符形式	含 义	ASCII 代码
\n	回车换行	10
\t	横向跳格 (Tab)	9
\b	退一格	8
\r	回车	13
\f	换页	12
\\	反斜线	92
\'	单引号	39
\"	双引号	34
\ddd	1~3 位八进制数所代表的字符	
\xhh	1~2 位十六进制数所代表的字符	

型常量。一个字符型常量在内存中以其对应的 ASCII 码值存放，例如在 ASCII 码字符集中，字符型常量 '0' ~ '9' 的 ASCII 码值是 48 ~ 57。显然 '0' 与数字 0 是不同的。

C 语言中还允许用一种特殊形式的字符型常量，即以反斜杠字符 “\” 开头的字符序列。前面用的 printf() 函数中的 '\n'，代表一个“回车换行”符。这类字符称为“转义字符”，意思是将反斜杠 “\” 后面的字符转换成另外的意义。转义字符及其含义如表 2-2 所示。